Postel

Network Working Group
Request for Comments #318
NIC #9348
Category: Telnet, Protocols
References: RFC 139, 158, and NIC 7104

Jon Postel
UCLA-NMC
3 April 72
Rcvd 28 March 72

At the October 1971 Network Working Group Meeting, I promised to promptly produce a document which clearly and succinctly specified and explained the Official Telnet Protocol. This document fails to meet any part of that promise. This document was not produced promptly. This document is neither clear nor succinct. There is no Official Telnet Protocol.

The following pages present my understanding of the ad hoc Telnet protocol. There are some who have serious questions about this protocol. The proposed changes to the protocol are given in Section IV.

Any comments should be promptly directed to me via the Network Information Center (Ident = JBP) or by phone (213) 825-2368 or by mail

Jon Postel
SPADE Group
3804 Boelter Hall
UCLA
Los Angeles, California 90024

Network Working Group
Request for Comments #318
NIC #9348

Jon Postel
UCLA-NMC
3 April 72

I. DEFINITION OF THE NETWORK VIRTUAL TERMINAL

The Network Virtual Terminal (NVT) is a bi-directional character device. The characters are represented by 8 bit codes. The NVT has no timing characteristics. The character codes 0 through 127 are the USASCII codes. (Note all code values are given in decimal.) The codes 128 through 255 are used for special control signals. The NVT is described as having a printer and a keyboard. The printer responds to incoming data and the keyboard produces outgoing data.

The Printer

The NVT printer has an unspecified carriage width (common values are 40, 72, 80, 120, 128, 132). The printer can produce representations of all 95 USASCII graphics (codes 32 through 126). Of the 33 USASCII control codes (0 through 31 and 127) the following 8 have specific meaning to the NVT printer.

Jon Postel
UCLA-NMC
3 April 72

NAME	CODE	MEANING
NULL (NUL)	0	A no operation.
BELL (BEL)	7	Produces an audible or visible signal.
Back Space (BS)	8	Backspaces the printer one character position.
Horizontal Tab (HT)	9	Moves the printer to next horizontal tab stop.
Line Feed (LF)	10	Moves the printer to next line (keeping the same horizontal position).
Vertical Tab (VT)	11	Moves the printer to the next vertical tab stop.
Form Feed (FF)	12	Moves the printer to the top of the next page.
Carriage Return (CR)	13	Moves the printer to the left margin of the current line.

The remaining USASCII codes (1 through 6, 14 through 31, and 127) do not cause the NVT printer to take any action.

The Keyboard

The NVT Keyboard has keys or key combinations or key sequences for generating all of the 128 USASCII codes. Note that although there are codes which have no effect on the NVT printer, the NVT Keyboard is capable of generating these codes.

Network Working Group
Request for Comments #318
NIC #9348

Jon Postel
UCLA-NMC
3 April 72

The End of the Line Convention

The end of a line of text shall be indicated by the character sequence Carriage Return Line Feed (CR LF). This convention applies to both the sending (Keyboard) and receiving (Printer) (virtual) mechanisms.

Break and Reverse Break

The Telnet control signals provide a BREAK signal which can be used to simulate the use of the break or attention or interrupt button found on most terminals. This signal has no effect on the NVT. When the BREAK Telnet control signal is used from server to user it is sometimes called "reverse break". Such a reverse break has no effect on the NVT.

Jon Postel
UCLA-NMC

II. DEFINITION OF TELNET PROTOCOL

The purpose of Telnet Protocol is to provide a standard method of interfacing terminals devices at one site to processes at another site.

The Telnet Protocol is built up from three major substructures, first the Initial Connection Protocol (ICP), second the Network Virtual Terminal (NVT), and third the Telnet control signals described herein.

Telnet user and server processes follow the ICP to establish connections. The term "Logger" has been associated with the set of processes in the serving system which respond to the ICP and perform the initial interactions e.g. obtain a name and password. The ICP is defined and the initial socket number and byte size parameters are defined in "Current Network Protocols" (NIC #7104).

The data transmitted between the user and server programs (and vice versa) is treated as a character stream with embedded control signals.

Note that all code values are given in decimal.

TELNET CONTROL SIGNALS

NAME	CODE	MEANING
DATA MARK	128	Used to mark a point in the data stream. Used in conjunction with INS. See SYNCH.

Jon Postel
UCLA-NMC

NAME	CODE	MEANING
BREAK	129	User-to-Server: Has the same meaning to the server as the "Break," Interrupt," or "Attention" button found on many terminals. Server-to-User: Has the same meaning to to use as the "reverse break" used with some terminals.
NOP	130	No Operation.
NO ECHO	131	User-to-Server: Asks the server not to return Echos of the transmitted data. Server-to-User: States that the server is not sending echos of the transmitted data. Sent only as a reply to ECHO or NO ECHO, or to end the hide your input.
ECHO	132	User-to-Server: Asks the server to send Echos of the transmitted data. Server-to-User: States that the server is sending echos of the transmitted data. Sent only as a reply to ECHO or NO ECHO.
Hide your input	133	The intention is that a server will send this signal to a user system which is echoing locally (to the user) when the user is about to type something secret (e.g. a password). In this case, the user system is to suppress local echoing or overprint the input (or something) until the server sends a NOECHO signal. In situations where the user system is not echoing locally, this signal must not be sent by the server.
INS	---	This is the "Interrupt on Send" signal, defined by the Host-to-Host protocol and implemented by the Network Control Program (NCP). See SYNCH.
SYNCH	---	This is a condition indicated by the combination of the DATA MARK and the INS. User-to-Server: The Server is to examine the input data stream looking for a DATA MARK signal; if a DATA MARK is found, the

Jon Postel
UCLA-NMC
3 April 72

NAME	CODE	MEANING
		server must not process further until an INS is received. If the server receives an INS, it is required to examine the data stream at once, taking any appropriate action on "break type" characters (e.g. etx, sub, BREAK), up to a DATA MARK signal and thereupon continue its normal processing. The passed over characters may be discarded.
		Server-to-User: If the user finds a DATA MARK in the data stream, it must wait for an INS. If the user receives an INS, it must examine and discard characters up to and including a DATA MARK.

DATA TYPES

Telnet normally deals in ASCII characters, but there are provisions for escaping to other code sets. If one of these escapes is used, it is undefined (here) whether or not the Telnet signals still have meaning or even how to return to the ASCII set: The Telnet signals used to indicate a change of code set are:

Code	Meaning
160	ASCII - Standard Telnet
161	Transparent
162	EBCDIC

USER TELNET SIGNALS

The following signals are to be abailable to the human user to cause the user Telnet to take the indicated action.

Jon Postel
UCLA-NMC
3 April 72

Transmit Now - Transmit all data entered and locally buffered now. Intended to be used with line mode operation.

Suppress end-of-line - Transmit all data entered and locally buffered now, and do not transmit the end-of-line immediately following this signal.

STANDARD TELNET IMPLEMENTATION

Using Site:

1) User is able to enter and transmit all ASCII codes.

2) User is able to cause the Telnet signals BREAK, SYNCH, ECHO and NOECHO to be transmitted.

3) Provides for the User Telnet signals, (e.g. Transmit Now).

4) Implements the CR LF end-of-line convention.

5) Provides local echo for local user terminals.

6) Correctly processes the Telnet signals BREAK, SYNCH, NOP, ECHO, NOECHO, and Hide Your Input received from ther server.

Serving Site:

1) Provides a mapping between ASCII and the local character set.

2) Correctly processes the Telnet signals BREAK, SYNCH, NOP, NOECHO and ECHO.

3) Implements the CR LF end-of-line convention.

4) Assumes the using site provides echoing. May provide a server echo mode.

MINIMUM TELNET IMPLEMENTATION

Using Site:

1) User must be able to enter and transmit all ASCII codes.

2) Ignore and delete all Telnet signals from the serving site.

Jon Postel
UCLA-NMC

3) Provide local echo for local user terminals.

4) Implements CR LF end-of-line convention.

5) Provide for the User Telnet signals.

Serving Site:

1) Provide a mapping between ASCII and the local character set.

2) Ignore and Delete all Telnet signals from the using site.

3) Assume the using site provides echoing.

4) Implements the CR LF end-of-line convention.

Jon Postel
UCLA-NMC

III. DISCUSSION OF TELNET PROTOCOL

The use of a standard, network-wide, intermediate representation of terminal code between sites is intended to eliminate the need for using and serving sites to keep information about the characteristics of each other's terminals and terminal handling conventions. This approach can be successful, but only if the user, the using site, and the serving site assume certain responsibilities.

1. The serving site must specify how the intermediate code will be mapped by it into the terminal codes that are expected at that site.

2. The user must be familiar with that mapping.

3. The using site must provide some means for the user to enter all of the intermediate codes, and as a convenience, special Telnet signals, as well as specify for the user how the signals from the serving site will be presented at the user terminal.

Other schemes were considered but rejected. For example, a proposal that the using site be responsible for translating to and from the code expected by the serving site was rejected since it required that the using site keep tables of all serving site codes and provide a mapping for each case. The information would require constant maintenance as new hosts were added to the network.

Character Set

Since it is not known how the current or future sites will specify the mapping between the network-wide standard code (7 bit ASCII in an 8 bit field) and the codes expected from their own terminals, it seems necessary to permit the user to cause transmission of every one of the

Jon Postel
UCLA-NMC
3 April 72

128 ASCII codes, plus (for full user power) selected signals (either of a Telnet control nature, or of a special terminal nature such as break or attention).

There was strong feeling about the importance of the user/system interface at the using site, but equally strong feeling that this problem is one of local implementation and should reflect the using site installation philosophy rather than be subject to network-wide standards. Some topics of consideration in this area are:

1. How to represent special graphics, not available at the using site, at the user's terminal.

2. Treatment of upper/lower case problem on upper case only devices.

   a. Representing lower-case output.

   b. Providing users with shift and shift lock signals.

3. Incorporating editing capability in Telnet.

4. Extending user options in Network mode not available to local users, e.g., hold output or kill print.

5. Permit users to specify how keyboard input is to be translated, e.g., let a character from the terminal cause a specified string to be sent by the user's Telnet.

The proposed solution to the Telnet Protocol problem seems to provide a mechanism for a minimum implementation while providing a basis for developing richer sets of protocol for present and future use in terminal applications, process-process communications, and use by other conventions to pass data or control information.

Network Working Group
Request for Comments #318
NIC #9348

Jon Postel
UCLA-NMC
3 April 72

The understanding that ASCII be used as a network-wide code has been established for some time. Its use in Telnet provided a problem with respect to the limitation of a maximum character set of 128. Some systems provide for more than this number in their operation, and therefore, as serving sites, cannot map on a one-for-one basis.

Each such serving site could probably provide a reasonably useful character set, including all system control signals, by mapping 128 of its codes and just not provide a network user access to the other codes. However, any character left out might later be used in a major application at that site as a special control signal. This could result in denying network users the facility offered by that application. Serving sites are, therefore, encouraged to provide a full mapping between the ASCII code and the code used on the serving system. This may require that the server specify two character ASCII sequences which map to single characters in the servers character set.

Notice that there are some ASCII codes which have no effect on the NVT printer. These codes must be transmitted over the network when output by the serving process or by entered by the user.

End of Line Convention

The representation of the end of a physical line at a terminal is implemented differently on different network hosts. For example, some use a return (or new line) key, the terminal hardware both returns the carriage or printer to start of line and feeds the paper to the next line. In other implementations, the user hits carriage return and the

hardware returns carriage while the software sends the terminal a line feed. The network-wide representation is carriage return followed by line feed. It represents the physical formatting that is being attempted, and is to be interpreted and appropriately translated by both using site and serving site.

Example: A Multics user is working, through the network, on some serving site host. In the course of the session the user has numerous occasions to hit New Line on his Model 37 TTY. Each time the Multics system is awakened by a New Line interrupt, the line of buffered characters is passed to Telnet where it is scanned for special characters. If none is found, carriage return followed by line feed is inserted where New Line was entered, and the line is turned over to the NCP for transmission. Correspondingly, when the Multics Telnet finds the carriage return line feed sequence in the data stream coming from the serving site, the two characters are replaced with the appropriate New Line code which is sent to the terminal.

Telnet defines the end of a line to be indicated by the ASCII character pair CR LR. Several of the real devices in the world have only a single new line (NL) function. Several of the computer systems have in some programs used the CR and LF functions to have semantic meaning larger than the format effect they provide. Further, several computer systems

allow the CR and LF functions to be used separately (e.g., such that a line may be overprinted). One problem, for those Telnet (user) programs required to map the NVT into a device which only has a NL function, is how is the CR LF to be dealt with. One solution is to examine the character following the CR. If an LF is found, then perform the NL function; if anything else is found then back space to the beginning of the line. Another problem is the case of a computer system which locally uses period, ".", to cause the new line function and which uses, in some programs, CR and LF for semantically significant operations. Suppose the user Telnet sends the sequence CR LF. Does this mean "new line" or the "CR operation" followed by the "LF operation"? A solution to this problem is to require that Telnet programs send a CR not intended to be part of a CR LF pair as a CR NUL pair. Then the receiving program can always hold a CR and examine the next character to determine if a new line function is intended. This solution is strongly recommended.

One other question arises here, "Is it permitted to send the Telnet signal NOP (code 130) between a CR and a LF when these are intended to signify new line?" The answer is "yes, the NOP signal may occur anywhere in the data stream."

Echoing

The decision to have the assumed condition for echo be that the using site will provide any echo necessary for its terminals was taken because of the difficulties faced by some installations that cannot

turn off their echo or that have terminals that print locally as a result of key strokes. Serving sites could take the position "have user turn echo off," but this seems an unnecessary burden on the user. In addition, some serving sites may choose not to supply any echo service, in which case the no echo assumption will supply a network-wide condition, while other assumptions would give a mixed starting condition.

The convention of using "ECHO," "NO ECHO" signals seems to fill both the requirements for dynamic echo control and for a minimum implementation of Telnet Protocol. Note that when the user request ECHO or NO ECHO the server replys by switching to the desired mode (and possibly returning the signal for the new mode), or by continuing in the current mode and returning the signal for the current mode. The server never spontaneously sends an ECHO or NO ECHO signal. Except that a NOECHO may be used to cancel a HIDE YOUR INPUT.

Hide Your Input

The HIDE YOUR INPUT signal presents some difficulty in that it is unclear how much is to be hidden. The server site usually knows how long the secret is but the user Telnet in general does not. Furthermore, if the user site cannot suppress the local echoing, there is a difficult implementation problem. One possibility is for the using site to overprint a full line with a mask, then have the user type his secret on the mask. If the secret were longer than one line, the use

of the mask should be repeated.

The use of HIDE YOUR INPUT can be avoided altogether by having the serving site send a mask (which it knows to be just long enough) on which the user is to type the secret information.

Example:

1. Default assumption is user site is echoing
2. Server-to-User: Password Please CR LF
3. Server-to-User: XXXXCRIIIICRMMMMCR NUL
4. User-to-Server: "password" CR LF
5. Server-to-User: Ready CR LF

Breaks and Attentions

There is a special control signal on some terminals that has no corresponding bit pattern in ASCII, but is transmitted by a special electrical signal. This control signal is Attn on a 2741 and Break on a Teletype. This signal is represented by the Telnet control signal BREAK. There is a corresponding control signal for use from serving sites to using sites for reverse break. Notice, however, that the NVT is a bi-directional character device, thus there is no need to "turn the line around".

Some systems treat the Break as an extra code available for use in conjunction with the data stream. For example, one system uses Break as a special editing code meaning "delete the current line to this point." In these cases, the code may simply be inserted in the

data stream with no special additonal action by the user.

Other systems use Break or Attn in a special interrupt fashion, to mean stop processing the application and give me the supervisor, or cancel the present job, etc. (Other systems which inspect input on a character at a time basis use normal characters for this purpose, such as <etx>.) In these cases, because of differences in the ways both serving and using sites operate, it is necessary to take a route in addition to the normal Telnet data stream to indicate that the special control signal is imbedded in the data stream.

Example--Problem:

The PDP-10 normally will, when it fills its input buffer, continue to accept characters from a terminal examining each to see if it is a control character, then act on it if it is or throw it away if it is not.

Since the Telnet server at the serving site is at the mercy of the NCP with respect to controlling the bunching, and therefore, arrival at the Telnet of bursts of characters, Telnet implementations might be expected to choke off flow to the buffers until they are ready to accept characters without throwing them away.

Under this condition, the serving process might be outputting to the using terminal, the input buffers at the server fill up, (with user generated characters) and <etx> gets stuck (at the user's site) in the data stream that has been choked off.

Jon Postel
UCLA-NMC
3 April 72

A similar problem could occur with Multics or some IBM system as a line at a time server. The user at a using site gets his process at the serving site into an output loop and wants to break the process without having to release his Telnet connection. The buffers clog the connection, transmission is choked off, and the <etx>, Break, or other user control signal gets stuck in the pipeline.

Example--Solution:

The user at the using site knows he is entering a special control signal (Break, Attn, <etx>, etc.) and follows it with a SYNCH. (The local instructions at using sites for accomplishing this may differ from site to site.)

User to Using Site Telnet:

Send SYNCH.

Using Site Telnet to Serving Site Telnet:

DATA MARK in Data Stream.

Using Site Telnet to Using Site NCP:

Send an INS.

Serving Site NCP to Telnet Server:

Interrupt "INS received".

Serving Site Telnet:

Examines the input data stream (looking for special control signals) until it sees DATA MARK then resumes normal handling.

Thus, depending on the server's local implementation to provide adequate service, a special handling of the data stream can be invoked whenever an INS is received in order to get to the special character. When it sees DATA MARK, it recognizes it as a synchronization point and knowing that the special character has been passed on, strips the DATA MARK from the data stream and returns to normal mode.

If the DATA MARK arrives before the INS, the serving site should not process the data stream further until an INS is received.

This approach to handling selected special characters or signals relieves the using Telnet processes from having to recognize the special serving site characters, as well as from having to know how the serving site wants to handle them. At the same time, the procedure requires only a minimum level of user understanding of the serving site. This seems appropriate, since the Telnet ASCII conventions are providing a Network Virtual Terminal, not a Network Virtual User.

Notice that the correct order is (1) special character or signal (e.g. BREAK or <etx>), then (2) SYNCH.

User Telnet Signals

The ability of the user to cause the using site Telnet to send any combination of ASCII characters in a string, and only that combination, is viewed as important to the user utility of the Telnet ASCII conventions. Because of this, some user sites may find it necessary

to provide special local Telnet signals from the human user to the using site Telnet.

Example:

A user on a line at a time system (Multics, System 360, GCOS, etc.), which require an end of line signal before processing the user's input, is working through the Network on a serving site that operates a character at a time. The application is a debugging aid that permits the user to type in "location =" to which it will respond with n where n represents the current contents of that location. The serving site process does not expect to see the "location =" followed by a carriage return line feed sequence. The user at the using site should be able to type in the location, follow it with a signal (to the user Telnet) to suppress the end of line convention, followed by the end of line signal, and expect the "location =" to be transmitted immediately without an end of line sequence being transmitted to the server.

Example:

In another case, a using site has decided that it is convenient to accumulate four characters at a time and transmit them to the serving site, unless an end of line signal is observed, in which case the end of line sequence is sent preceded by whatever number of characters have been accumulated (presumably three or less). In the same debugging application, the address is such

that the end does not correspond with the four character buffer demarcation. The user should have the ability to enter a code for "transmit immediately" in place of the end of line signal in order to preserve neat formatting, and expect the address to be sent to the serving site.

Telnet Signals have been discussed and those introduced to date are probably sufficient for an implementation of Telnet ASCII convention.

Terminology

ASCII - The USASCII character set as defined in NIC #7104. In Telnet Protocol, where eight bit codes are used the lower half of the code set is defined to be ASCII.

echoing - The display of a character entered is called echoing. There are two modes in which this happens. If the device used to enter characters displays the character before (or as) it transmits the character to the computer the echoing mode is called "local echo." If, on the other hand, the device transmits the entered character to the computer without displaying it and the computer then transmits a character to the device for the echo display, this echoing mode is called "remote echo."

Jon Postel
UCLA-NMC
3 April 72

character mode - In this mode of operation Telnet transmits each character as soon as possible. Generally speaking, character mode is used when all of the using terminal, using system, and serving system are operating in a remote echo mode. The echos to the user entered characters are transmitted from the serving system (i.e., over the network).

line mode - In this mode of operation Telnet transmits groups of characters which constitute lines. Generally speaking, this mode is used when one or more of the using terminal, using system, or serving system is operating in a local echo mode. The echos to the user entered characters are not transmitted over the network).

full duplex - This term indicates a transmission procedure using a four wire connection, which permits simultaneous transmission in both directions.

half duplex - This term indicates a transmission procedure using a two wire connection, which requires that data be transmitted in only one direction at a time.

Note that half duplex devices usually are also local echo but that full duplex devices may be either local echo or remote echo.

Jon Postel
UCLA- NMC

IV. PROPOSED CHANGES TO TELNET PROTOCOL

The changes suggested here are not my ideas, thus the presentation may be faulty. I welcome RFC or other communication suggesting other changes or better arguments for and against these changes.

Echoing

It is proposed to delete from Telnet the control signals ECHO, NOECHO, and HIDE YOUR INPUT. For Server systems which do not provide echoing, these commands are useless. For server systems which do provide echoing experience has shown that the control is most effectively provided by server system commands.

Data Types

It is proposed to delete all mention of data types from Telnet. Either the character stream is ASCII or its not a Telnet communication. If it is really necessary to change the data type, a command in ASCII could be sent in the data stream.

Minimum Implementation

It is proposed that the minimum implementation require the user Telnet to allow the user to send and the server Telnet to correctly process all the Telnet control signals.

Network Working Group
Request for Comments #318
NIC #9348

Jon Postel
UCLA-NMC
3 April 72

The work on Telnet Protocol has involved many people. This document is taken from RFC's #139 and #158 by Tom O'Sullivan. Others who have served on committees are:

Bob Bressler	MIT-DMCG
Will Crowther	BBN
Bob Long	SDC
Alex McKenzie	BBN
John Melvin	SRI-ARC
Bob Metcalfe	MIT-DMCG
Ed Meyer	MIT-Multics
Tom O'Sullivan	Raytheon
Mike Padlipsky	MIT-Multics
Jon Postel	UCLA-NMC
Bob Sundberg	Harvard
Joel Winett	LL
Steve Wolfe	UCLA-CCN